PROJECT REPORT

A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITIONSYSTEM

Submitted by

PNT2022TMID39323

HARINI T	**190801060**
KEERTHANA H	**190801087**
MADHUMITHA S	**190801098**
NANDHITHA P	**190801102**

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 PROJECT OVERVIEW

Recognition is identifying or distinguishing a thing or an individual from the past experiences or learning. Similarly, Digit Recognition is nothing but recognizing or identifying the digits in any document. Digit recognition framework is simply the working of a machine to prepare itself or interpret the digits. Handwritten Digit Recognition is the manually written digitsfrom various sources like messages, bank cheques, papers, pictures and so forth and in varioussituations for web based handwritting recognition on PC tablets, identifying number plates of vehicle, handling bank cheques, digits entered in any forms etc.

Machine Learning provides various methods through which human efforts can be reduced in recognizing the manually written digits. Deep Learning is a machine learning methodthat trains computers to do what easily falls into place for people: learning through examples. With the utilization of deep learning methods, human attempts can be diminished in perceiving,learning, recognizing and in a lot more regions. Using deep learning, the computer learns to carry out classification works from pictures or contents from any document. Deep Learning models can accomplish state-of-art accuracy, beyond the human level performance. The digit recognition model uses large datasets in order to recognize digits from distinctive sources.

The task of handwritten digit recognition, using classifier, has extraordinary significance and use such as-online digit recognition on PC tablets, recognize zip codes on mail, processingbank check amounts, numeric sections in structures filled up by hand and so on. There are diverse challenges faced while attempting tob solve this problem. The handwritten digits are not always of the same size, thickness, or orientaion and position relative to the margins. The main objective was to actualize a pattern charecterization method to perceive the handwritten digits provided in the MINIST data set of images of handwritten digits(0-9).

1.2 PURPOSE

The aim of a handwritten digit recognition system **is to convert handwritten digits intomachine readable formats.** The main objective of this work is to ensure effective and reliable approaches for recognition of handwritten digits and make banking opreations easier and errorfree. Handwritten character recognition is one of the practically important issue in pattern recognition applications. The applications of digit recognition includes in **postal mail sorting, bank check processing, form data entry, etc.**

CHAPTER 2

LITERATURE SURVEY

2.1 EXISTING PROBLEM

The fundamental problem with haandwritten digit reconition is that handwritten digits donot always have the same size, width, orientation, and margins since they vary from person to person. Additionally, there would be issues with identifying the numbers because of similaritiesbetween numerals like 1 and 7,5 and 6,3 and 8,2 and 5,2 and 7,etc. Finally, the individuality and variation of each individual's handwrting influence the structure and appearance of the digits.

2.2 REFERENCES

Handwritten Digit Recognition using Machine and Deep Learning Algorithms (2021)
Ritik Dixit, Rishika Kushwah, Samay Pashine

This paper provides a reasonable understanding of machine learning and deep learning algorithms like SVM, CNN, and MLP for handwritten digit recognition. It further more gives information about which algorithm is efficient in performing the task of digit recognition. In further sections of this paper, we will be discusing the related work that has been done in thuis field followed by the methodology and implementation of all the three algorithms for the fairer understanding of them. Next, it presents the conclusion and result bolstered by the work we have done in this paper. Moreover, it will also give you some potential future enhancements that can be done in this field.

Handwritten Digit Recognition using Convelutional Neural Networks (2022)
Sachin S Panchal

This paper think about the exhibition of Convolution Neural Network (CNN). Results demonstrate that CNN classifier beat over the Neural Network with critical improved computational effectiveness without relinquishing execution. Handwritten digit recognition canbe performed using the convolutional neural network from machine learning. using the MNIST(Modified National Institute of Standards and Techonologies) database and compiling with the CNN gives the basic structure of my project development. this work is to provide realistic and robust approaches for confirming translated mathematics by examining several current plan models.

Hybrid CNN-SVM classifier for Handwritten Digit Recognition (2020)
Savitha Ahlawata, Amit Choudharyb

The aim of this paper is to develop a hybrid model of a powerful Convolutional Neural Network (CNN) and supports Vector Machine (SVM) for recognition of handwritten digits from MNIST dataset. The proposed hybrid model combines the key properties of both the classifiers. In the proposed hybrid model, CNN works as an automatic feature extractor and SVM works as a binary classifier. The MNIST dataset of handwritten digits used for training and testing the algorithm adopted in the proposed model.

Effective Handwritten Digit Recognition using Deep convolutoinal Neural Network (2020)
Yellapragada SS Bharadwaj, Rajaram P, Sriram V.P, Sudhakar S, Kolla Bhanu Prakash

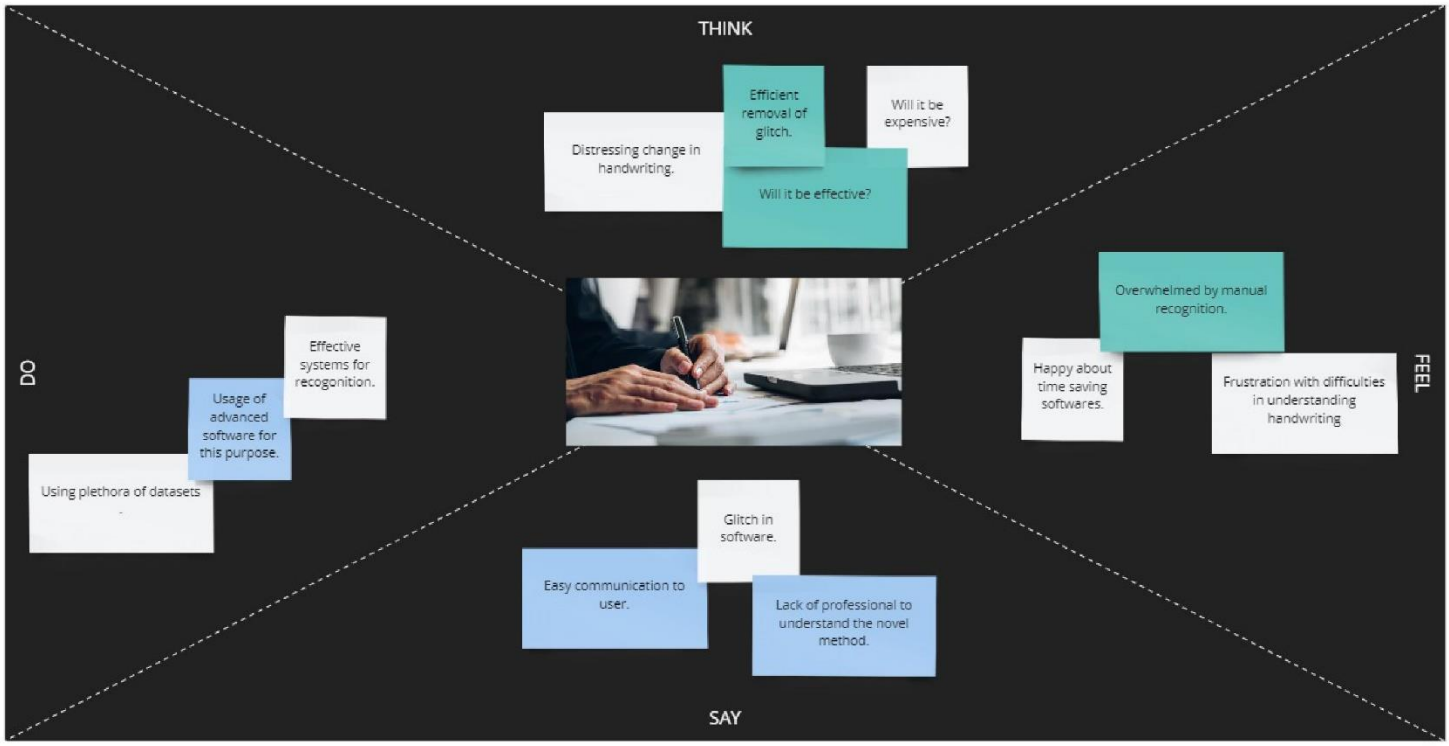
A simple neural network approach towards handwritten digit recognition using convolutional. With machine learning algorithms like KNN, SVM/SOM, recognizing digits is considered as one of the unsolvable tasks due to its distinctiveness in the style of writting. In this peper, convolutional Neural Networks are implemented with an MNIST dataset of 70000 digits with 250 distinct forms of writtings.

2.3 PROBLEM DEFINITION

The handwritten digits are not always of the same size, width, orientation and justified tomargins as they differ from writing of person to person, so the general problem would be while classifying the digits due to the similarity between digits such as 1 and 7, 5 and 6, 3 and 8,2 and 5,2 and 7,etc. This problem is faced more when many people write a single digit with a variety of different handwritings. Lastly, the uniqueness and varity in the handwritingds of different individuals also influence the formation and appearance of the digits.

CHAPTER 3
IDEATION AND PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS



Pain
1. Inefficient usage of software
2. Difficulty in navigation
3. Absence of skill to use technology

Gain
1. Efficient method to understand handwriting
2. Consumes time
3. Faster technology

3.2 IDEATION & BRAINSTROMING

Handwriting recognition has become a widely researched as every person has their own individual handwriting. Recognising and identifying this has thus become very important and useful software. Digit Recognition is recognizing or identifying the digits in any document. In digit recognition framework, the system recognises or identifies digits in a document. Handwritten Digit Recognition is the role of a computer to interpret the manually written digits from the sources like messages, papers, recognize license plates ,bank checks, numbers in any form, etc.

VARIOUS IDEAS INCLUDE:

1. Feature extraction for recognising various different handwriting structures.
2. CNN (Convolution neural networking) to execute an efficient system for handwriting recognition.
3. High accuracy machine learning techniques for this purpose.
4. MNIST dataset for training the various image processing systems.

Creating an effective yet flexible image recognition system for the same purpose is the objective. By modifying existing systems by using the above methods, this can be achieved.

3.3 PROPOSED SOLUTION

PROBLEM STATEMENT

In this digital world, everything including documents, notes is kept in digital form. The requirement of converting these digital documents into processed information is in demand. Handwriting recognition is one of the compelling research works going on because every individual in this world has their own style of writing. It is the capability of the computer to identify and understand handwritten digits or characters automatically. Because of the progressin the field of science and technology, everything is being digitalized to reduce human effort. Hence, there comes a need for handwritten digit recognition in many real-time applications.

SOLUTION DESCRIPTION

Convolutional Neural Networks (CNN) becomes one of the most appealing approachesand has been an ultimate factor in a variety of recent success and challengingmachine learning applications such as challenge ImageNet objectDetection image segmentation and face recognition. Therefore, we choose CNN for our challenging tasks of image classification. We can use it for handwriting digits recognition which is one of high academic and business transactions. There are many applications of handwriting digit recognition in our real life purposes. Precisely, we can use it in banks for reading checks, post offices for sorting letter, and many other related works.

MNIST database

The MNIST database (Modified National Institute of Standards and Technology database)is a handwritten digits dataset. We can use it for training various image processing systems[11]. The database is also widely used for training and testing in the fieldof machine learning. It has 60,000 training and 10,000 testing examples. Each image has fixed size. The images are of size 28*28 pixels. It is a database for people who want to try learning techniques and pattern recognition methods on real-world data while spending minimal efforts on pre-processing and formatting. We will use this database in our experiment.

Convolutional Neural Networks

Convolutional neural networks are deep artificial neural networks.We can use it to classify images (e.g., name what they see), cluster them by similarity (photo search)and perform object recognition within scenes. It can be used to identify faces, individuals, street signs, tumors, platypuses and many other aspects of visual data. The convolutional layer is the core building block of a CNN. The layer's parameters consist of a set of learnable filters (or kernels) which have a small receptive field but extend through the full depth of the input volume. During the forward pass, each filter is convolved across the width and height of the input volume, computing the dot product, and producing a 2-dimensional activation map of that filter. As a result, the network learns when they see some specific type of feature at some spatial position in the input. Then the activation maps are fed into a down sampling layer, and like convolutions, this methodis applied one patch at a time.CNN has also fully connected layer that classifies output with one label per node.

Methodology

Deep Learning has emerged as a central tool for self-perception problems like understanding images, a voice from humans,robots exploring the world. We aim to implement the concept of Convolutional Neural Network for digit recognition. Understanding CNN and applying it to the handwritten digit recognition system is the target of the proposed model. Convolutional Neural Network extracts the features maps from the 2D images. Then it can classify the images using the features maps. The convolutional neural network considers themapping of image pixels with the neighbourhood space rather than having a fully connected layer of neurons.The convolutional neural network is a powerful tool in signal and image processing. Even in the fields of computer vision such as handwriting recognition, natural object classification, and segmentation, CNN has been much better tool compared to all other previously implemented tools. The broader aim may be to develop a machine learning model that could recognize people's handwriting.

Libraries Required

Make sure that the following libraries are installed on your working machine before proceeding further

- Keras
- Tensorflow
- Numpy

NOVELTY

The handwritten digit recognition is the capability of computer applications to recognize the human handwritten digits. It is a hard task for the machine because handwritten digits are not perfect and can be made with many different shapes and sizes. The handwrittendigit recognition system is a way to tackle this problem which uses the image of a digit and recognizes the digit present in the image. Handwritten Digit Recognition is the capability of a computer to fete the mortal handwritten integers from different sources like images, papers, touch defenses, etc. And classify them into 10 predefined classes(0-9).

This is the existing method along with this we add some features to make our projectunique among them. We create a model that recognize multiple-digits present in the image ina offline mode. The future scope of this method is, we can recognize number in License plate, bank cheques and postal mail sorting.

We get a predicted result in two manner one is, the recognized digits is showed in the interface and the another manner is, we can get a predicted result through voice mode. This means the model tell the multiple-digit in voice mode. This feature helps the old age people they are difficult in understanding handwritten digits, blind people and who contain eye sight issues.

SOCIAL IMPACT

Digit recognition plays an role in the modern world. 'Digits' are a part of our everyday life, be it License plate on our cars or bike, the price of a product, speed limit plate on our carsor bike, the price of a product, speed limit on a road, or details associated with a bank account. In the case of a text which is unclear, it is easier to guess the digits in comparison to the alphabets.

Machine Learning and Deep Learning are reducing human efforts in almost every field. Moreover, a solution achieved using ML and DL can power various applications at the same time, thereby reducing human effort and increasing the flexibility to use the solution. One such solution is a handwritten digit recognition system that can be used in postal mail sorting, bank check processing, form data entry, etc.

- It can solve more complex problems and makes humans job easier. This type of system can be widely use in the world to recognize zip code or postal code for mailsorting.
- In banking sector too where more handwritten numbers are involved like accountnumber, figure of cash and checks
- Inbanking system it is used to recognizing written digits on cash deposit/withdrawal/and other

transaction which is also able to recognize the handwritten account number and amount on the cash deposit slip and thus automatethe cash deposit process at the bank counter.

- Postal department and courier services can easily find the digitswritten.
- Old people who will have eye sight issues with handwritten digits.

BUSINESS MODEL

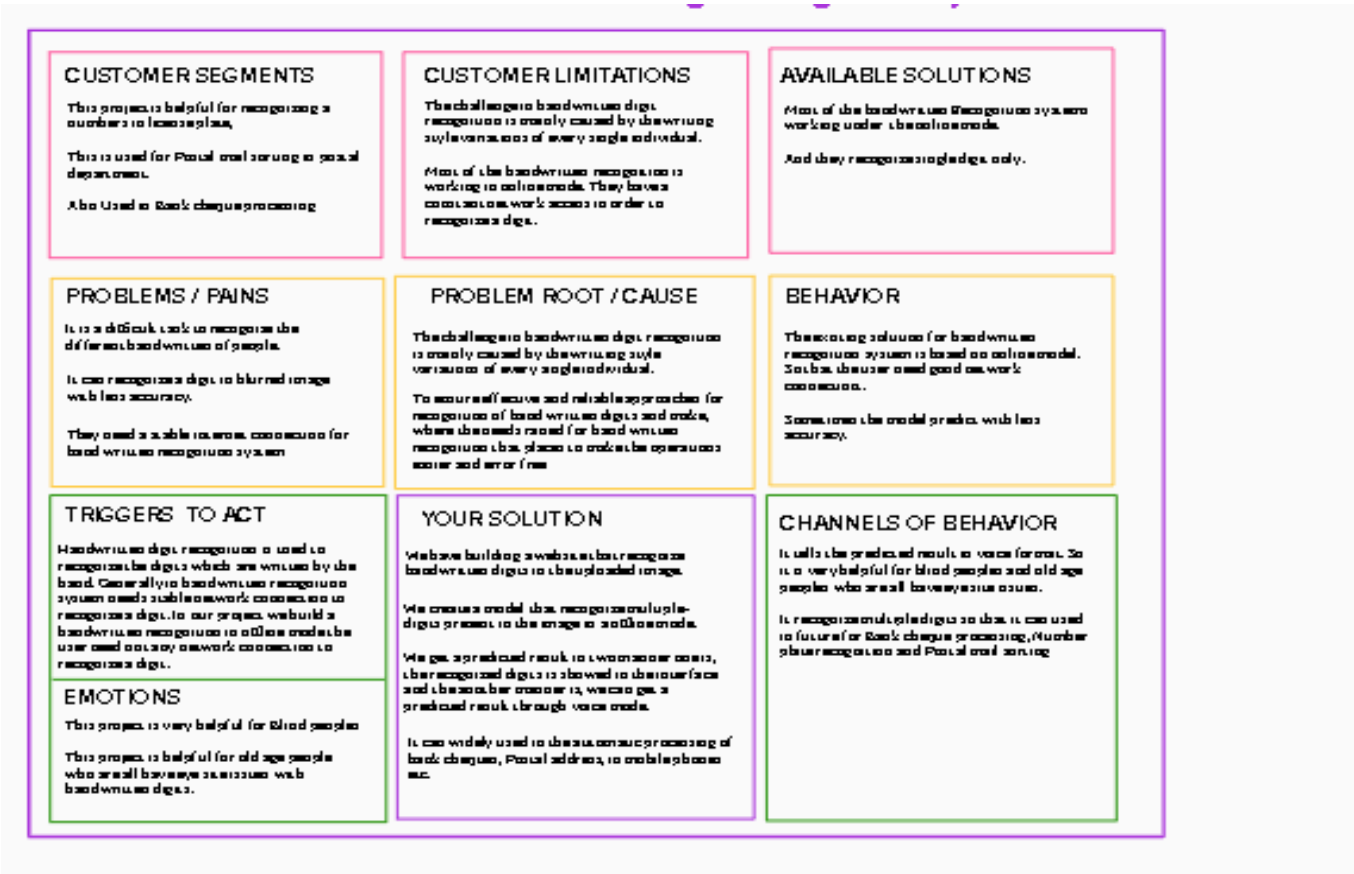
Digit recognition plays an important roles in many places. It is independent of environment, while using the recognizer we don't need the network. The benefits of hand written digit recognizer is high. In banking sector, it is very useful. It is used to recognize the account number, figure of cash and checks. It is also used to recognize the written digits on cash, deposit /withdrawal. So, the requirement of manpower is less, because the machine done the work of bank employees. So, we can earn the profit by using the hand written digit recognizer.

The hand written digit recognizer is used in postal department it recognize the pincodes, zip codes. It reduce the required amount of man power. It gives more accuracy. By using this handwritten digit recognizer postal department earn the profit /revenue. Because of the required man power is less and it also gives the better accuracy. This recognizer is widely used in many places, such as mail sorting, form data entry etc... Whenever the demand arisesfor digital recognizer, on that place we can use it and earn the profit by this handwritten digit recognizer.

SCALABILITY OF SOLUTION

To make the path toward digitalization clearer by providing high accuracy and faster computational for recognizing the handwritten digits. The present Neural network as classifier, MNIST as dataset with suitable parameters for training and testing and frame work for hand written digit recognition .The aforementioned system successfully imparts accuracy up to 99.20% which is higher than formally proposed schemes. In addition, the proposed system reduces computational time significantly for training and testing due to which algorithm become efficient earning parameters in designing a CNN that leads us to reach a new absolute record in classifying MNIST handwritten. Thus the CNN architecture is proposed in order to achieve accuracy even better than that of ensemble architectures, along with reduced operational complexity and cost. Moreover, we also present an appropriate combination of l digits. We carried out extensive experiments and achieved a recognition accuracy of 99.87% for a MNIST dataset.

3.4 PROBLEM SOLUTION FIT



CHAPTER 4

REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS

FR No.	Functional Requirement(Epic)	Sub Requirement(Story/Sub-Task)
FR-1	Getting the handwritten digit input	The handwritten digit is obtained as input from the user as an image uploading or writing on the canvas.
FR-2	Data preprocessing	Upgrades the image to make it ready for segmentation, by performing on asks on the input image.
FR-3	Segmentation & Feature Extraction	Segment the MNIST dataset images using edge detection technique and removered unhandy from the data
FR-4	Classification and Recognition	Passing the feature vectors as individual input to the classifiers or neural networks such as CNN.
FR-5	Prediction	The deep learning model is trained and tested using the MNIST dataset, with accuracy>90%
FR-6	Evaluation	Ensure that the digit is correctly recognized by the model and produces accurate output.

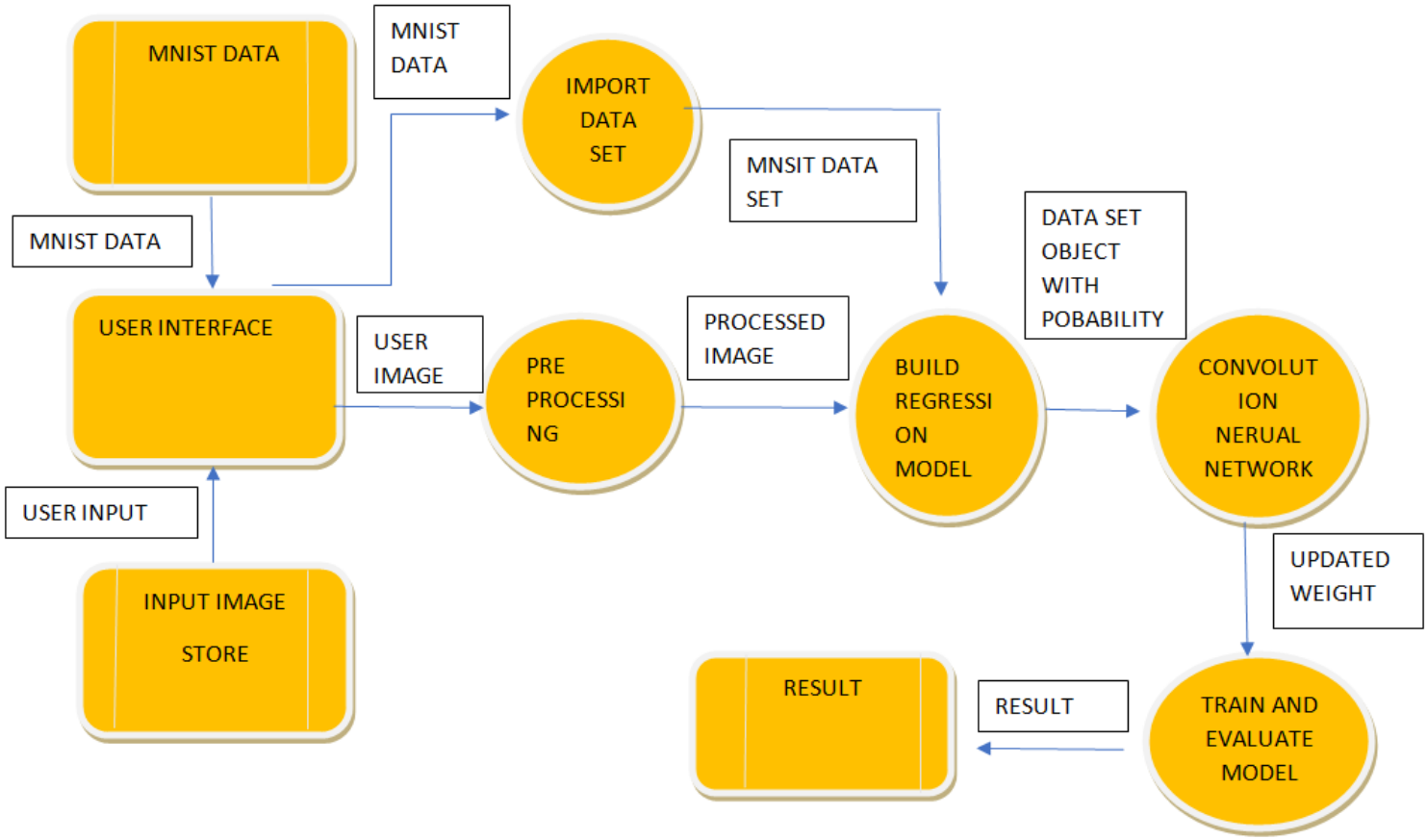
NON-FUNCTIONAL REQUIREMENTS

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	To identify and understand handwritten digits automatically, with high accuracy.
NFR-2	**Security**	Ensures security, since uploaded images are not stored in any database
NFR-3	**Reliability**	User-friendly webi nterface for the system.Process confidential information without dataleakage.

NFR-4	**Performance**	High, since artificial neural networks are used to train the images and build deep learning model. Fast prediction using CNN algorithm.
NFR-5	**Availability**	Using web application ,any one can easily access the system, making It highly available for we band mobile browsers.
NFR-6	**Scalability**	Performs well even if the count of input handwriting increased, since MNIST dataset is used for recognition process. Low time consumption.

CHAPTER 5
PROJECT DESIGN

5.1 DATA FLOW DIAGRAM

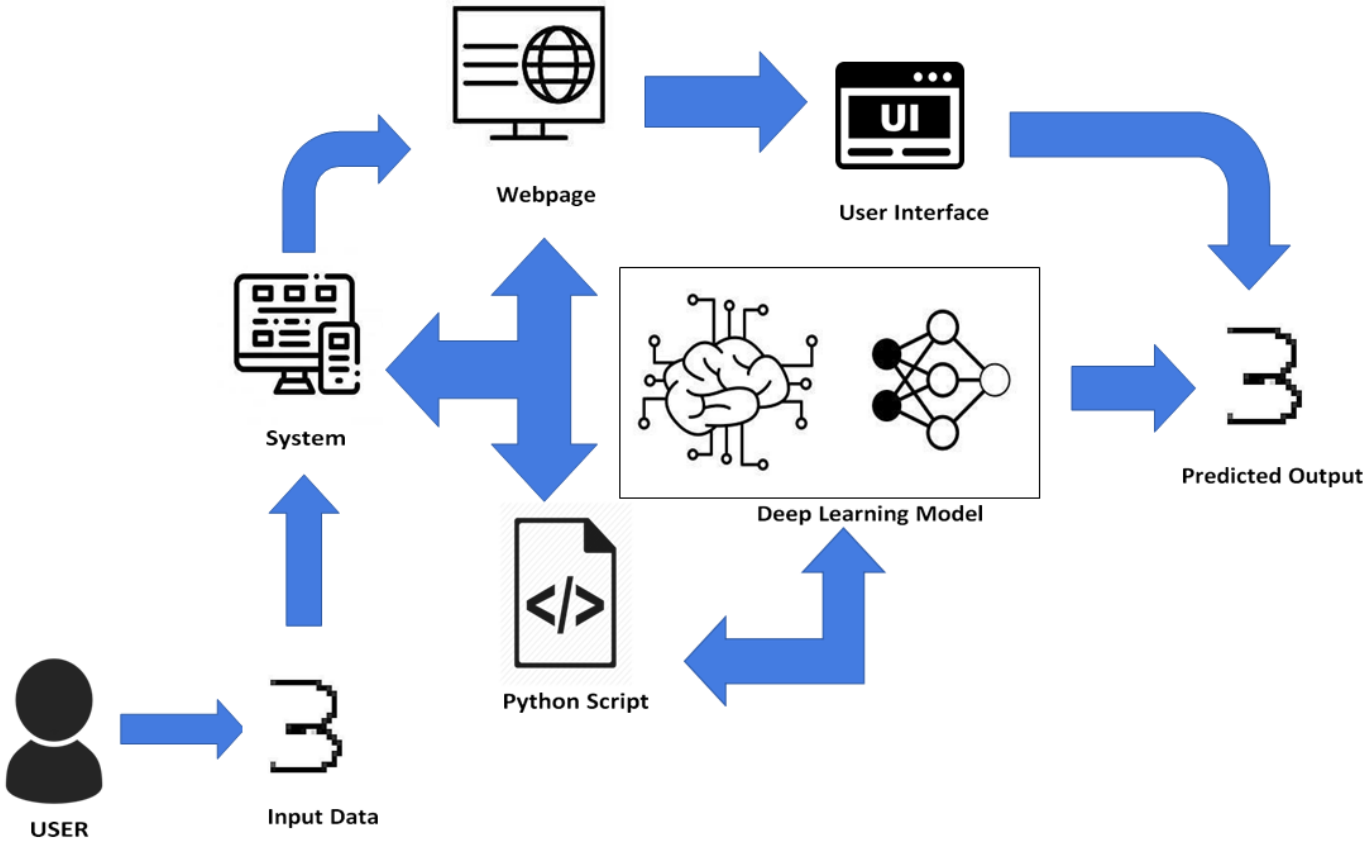


5.2 SOLUTION & TECNICAL ARCHITECTURE

ABSTRACT

In this digital world, everything including documents, notes is kept in digital form. The requirement of converting these digital documents into processed information is in demand. Handwriting recognition is one of the compelling research works going on because every individual in this world has their own style of writing. It is the capability of the computer to identify and understand handwritten digits or characters automatically. Because of the progressing the field of science and technology, everything is being digitalized to reduce human effort. Hence, there comes a need for handwritten digit recognition in many real-time applications. We create a model that recognize multiple-digits present in the image in offlinemode. The future scope of this method is, we can recognize number in License plate, bank cheques and postal mail sorting. We get a predicted result in two manner one is, the recognized digits is showed in the interface and the another manner is, we can get a predicted result through voice mode. This means the model tell the multiple-digit in voice mode.

TECHINICAL ARICHITECTURE



METHODOLOGY

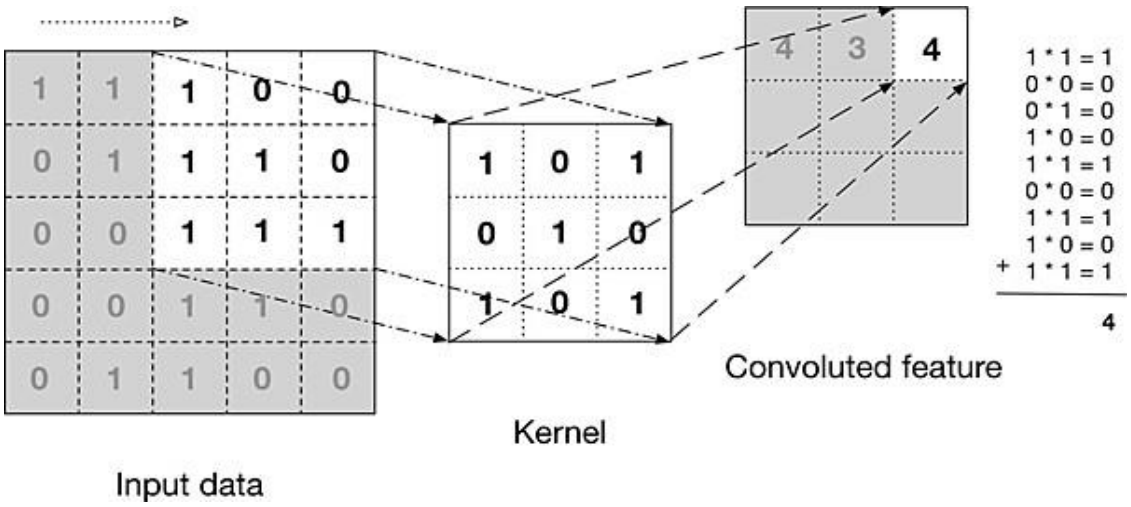
Deep Learning has emergedas a central tool for self-perception problemslike understanding images,a voice from humans, robots exploring the world. We aim to implement the concept of Convolutional Neural Network for digit recognition. Understanding CNN and applying it to the handwritten digit recognition system is the target of the proposed model.

Convolutional Neural Network extracts the features maps from the 2D images. Then it can classify the images using the features maps. The convolutional neural network considers the mapping of image pixels with the neighbourhood space rather than having a fully connected layer of neurons. The convolutional neural network is a powerful tool in signal and image processing. Even in the fields of computer vision such as handwriting recognition, natural object classification, and segmentation, CNN has been much better tool compared to all other previously implemented tools. The broader aim may be to develop a machine learning modelthat could recognize people's handwriting.

CONVOLUTIONAL NEURAL NETWORK

Convolutional neural networks are deep artificial neural networks. We can use it to classify images (e.g., name what they see), cluster them by similarity (photo search) and perform object recognition within scenes. It can be used to identify faces, individuals, street signs, platypuses and many other aspects of visual data. The convolutional layer is the core building block of a CNN. The layer's parameters consist of a set of learnable filters(or kernels) which have a small receptive field but extend through the full depth of the input volume. During the forward pass, each filter is convolved across the width and height of the input volume, computing the dot product, and producing a 2-dimensional activation map of that filter. As a result, the network learns when they see some specific type of feature at some spatial position in the input. Then the activation maps are fed into a down sampling layer, andlike convolutions, this method is applied one patch at a time. CNN has also fully connected layer that classifies output with one label per node. Convolutional neural networks are composed of multiple layers of artificial neurons. Artificial neurons, a rough imitation of their biological counterparts, are mathematical functions that calculate the weighted sum of multiple inputs and outputs an activation value. When you input an image in a Conv Net, each layer generates several activation functions that are passed on to the next layer.

The first layer usually extracts basic features such as horizontal or diagonal edges. This output is passed on to the next layer which detects more complex features such as corners or combinational edges. As we move deeper into the network it can identify even more complex features such as objects, faces, etc.



STEPS TO IMPLEMENT HANDWRITTEN DIGIT RECOGNITION

Import the libraries and load the dataset

First, we are going to import all the modules that we are going to need for training our model. The Keras library already contains some datasets and MNIST is one of them. So we can easilyimport the dataset and start working with it. The **mnist.load_data()** method returns us the training data, its labels and also the testing data and its labels

Pre-process the data

The image data cannot be fed directly into the model so we need to **perform some operationsand process the data** to make it ready for our neural network. The dimension of the training data is (60000,28,28). The CNN model will require one more dimension so we reshape the matrix to shape (60000,28,28,1).

Create the model

Now we will **create our CNN model** in Python data science project. A CNN model generally consists of convolutional and pooling layers.It works better for data that are represented as grid structures, this is the reason why CNN works well for image classification problems. The dropout layer is used to deactivate some of the neurons and while training, it reduces offer fitting of the model. We will then compile the model with Adadeltaoptimize.

Train the model

The **model.fit() function** of Keras will start the training of the model. It **takes the training data, validation data, epochs, and batch size.** It takes some time to train the model. After training, we save the weights and model definition in the 'mnist.h5' file.

Evaluate the model

We have 10,000 images in our dataset which will be used to **evaluate how good our model works**. The testing data was not involved in the training of the data therefore, it is newdata for our model.The MNIST dataset is well balanced so we can get around99% accuracy.

Create GUI to predict digits

Now for the GUI, we have created a new file in which we **build an interactive window todraw digits on canvas** and with a button, we can recognize the digit. The Tkinter library comes in the Python standard library. We have created a function **predict_digit()** that takes the imageas input and then uses the trained model to predict the digit. Then we **create the App class** which is responsible for building the GUI for our app. We create a canvas where we can drawby capturing the mouse event and with a button, we trigger the predict_digit() function and display the results.

DATASET

In this paper, we used the MNIST database consisting of offline handwritten digits ranging from 0-9. The database was constructed from Special Database 3 (SD-3) and SpecialDatabase1 (SD-1) that contain binary images of handwritten digits.SD-3 was collected among Census Bureau employees, while SD-1 was collected among high school students. For the results to be independent of both datasets, MNIST dataset was built by mixing NIST SD SD-3.The total number of digit image samples (70,000), the total number for training (60,000) and testing (10,000), and the subtotal number for each digit are shown in table 1. Each digit is a gray-level fixed-size image with a size of 28 x 28 (or 784 pixels) in total as the features.

In this paper, we used the MNIST database consisting of offline handwritten digits ranging from 0-9. The database was constructed from Special Database 3 (SD-3) and SpecialDatabase1 (SD-1) that contain binary images of handwritten digits.SD-3 was collected amongCensus Bureau employees,

while SD-1 was collected among high school students. For the results to be independent of both datasets, MNIST dataset was built by mixing NIST SD SD-3.The total number of digit image samples (70,000), the total number for training (60,000)and testing (10,000), and the subtotal number for each digit are shown in table 1. Each digit is a gray-level fixed-size image with a size of 28 x 28 (or 784 pixels) in total as the features.

5.3 USER STORIES

User Type	Functional Requirement	User Story Number	User Story/ Task	Acceptance Criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-2
		USN-3	As a user, I can register for the application through Gmail or Facebook	I can register & access the dashboard with Facebook Login	Medium	Sprint-2
	Login	USN-4	As a user, I can log into the application by entering email & password	I can login to the application	High	Sprint-1
	Dashboard	USN-5	Go to dashboard and refer the content about our project	I can read instructions also and the home page is user-friendly.	Low	Sprint-1
	Upload Image	USN-6	As a user, I can able to input the images of digital documents to the application	As a user, I can able to input the images of digital documents to the application	High	Sprint-3
	Predict	USN-7	As a user I can able to get the recognised digit as output from the images of digital documents or images	I can access the recognized digits from digital document or images	High	Sprint-3

		USN-8	As a user, I will train and test the input to get the maximum accuracy of output.	I can able to train and test the application until it gets maximum accuracy of the result.	Medium	Sprint-4
Customer (Web user)	Login	USN-9	As a user, I can use the application by entering my email, password.	I can access my account	Medium	Sprint-4
Customer Care Executive	Dashboard	USN-10	upload the image	Recognize and get the output	High	Sprint-1
Administrator	Security	USN-11	updated the features	checking the security	Medium	Sprint-1

CHAPTER 6
PROJECT PLANNING AND SCHEDULING

6.1 SPRINT PLANNING AND ESTAMATION

Sprint	Functional requirements	User story number	User story/Task	Story points	Priority	Team members
Sprint-1	Data collection and pre-processing	USN-1	As a user, I can upload any kind image with the pre-processing step is involved in it	10	High	Keerthana H,Harini T
		USN-2	As a user, I can upload the image in any resolution	10	High	Nandhitha P, Keerthna H
Sprint-2	Building the machine learning model	USN-3	As a user, I will get a application with ML model which provides high accuracy of recognized handwritten digit	10	Medium	Nandhitha P, Harini T

		USN-4	As a user, I can pass the handwritten digit image for recognize the digits	10	Medium	Madhumitha S, Harini T
Sprint-3	Application	USN-5	As a user, I	5	High	A
	building		access and read the instruction			Nandhitha P, Madhumitha S
		USN-6	As a user, I can use the next button to navigate to the next page	2	Medium	Nandhitha P, keerthana H, Harini T
		USN-7	As a user, I can upload the handwritten digit image in the website	6	High	Madhumktha S, Keerthana H, Nandhitha P
		USN-8	As a user, I can see the predicted / recognized digits in the application	7	Medium	Harini T, Madhumitha S

Sprint-4	Train and deployment of modelin IBM cloud	USN-9	As a user, I can access the web application and make the use of the product from anywhere	20	High	Harini T, Keerthana H, Madhumitha S

6.2 SPRINT DELIVARY SCHEDULE

SPRINT	TOTAL STORY POINTS	DURATION	SPRINT START DATE	SPRINT END DATE	STORY POINTS COMPLETED (AS ON PLANNED DATE)	SPRINT RELEASE DATE(DATEACTUAL)
Sprint-1	20	6 days	24 oct 2022	29 oct 2022	20	29 oct 2022
Sprint-2	20	6 days	31 oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

CHAPTER 7

CODING AND SOLUTIONING

```python
#import the packages
from flask import Flask, render_template,request
import tensorflow as tf
import numpy as np
from PIL import Image
from tensorflow.keras.models import load_model

app=Flask(__name__,template_folder='template')

#index page load
@app.route("/")
@app.route("/index")
def home():
    return render_template("index.html")

#load the  main page
@app.route("/main",methods=['GET'])
def main():
    return render_template("main.html")

# get the image in app.py and predict the image
@app.route("/main",methods=['POST'])
def upload():

# already trained model as been save this code load the model here
    model = load_model('model/mnistCNN.h5')
```

```python
#image is preproceesing and predicte the image
    imagefile = request.files['imagefile']
    if imagefile:
        img = Image.open(request.files['imagefile'].stream).convert("L")
        img = img.resize((28,28))
        image = np.array(img)
        image = image.reshape(1,28,28,1)
        prid = model.predict(image)
        prid =np.argmax(prid,axis= 1)

# predict the image and if condition using load the html page
        if (int(prid) == 0):
            return render_template("zero.html")
        if (int(prid)== 1):
            return render_template("one.html")
        if (int(prid)== 2):
            return render_template("two.html")
        if (int(prid)== 3):
            return render_template("three.html")
        if (int(prid)== 4):
            return render_template("four.html")
        if (int(prid)== 5):
            return render_template("five.html")
        if (int(prid)== 6):
            return render_template("six.html")
        if (int(prid)== 7):
            return render_template("seven.html")
        if (int(prid)== 8):
            return render_template("eight.html")
        if (int(prid)== 9):
            return render_template("nine.html")
    else:
        return render_template("main.html")
if __name__=='__main__':
    app.run(debug=True)
```

CHAPTER 8

TESTING

8.1 TEST CASES

Test case ID	Feature Type	Component	Test scenario	Pre Requisite	Steps to execute	Expected Result	Actual Result	status
HP_TC_O O1	UI	Home Page	Verify UI elements in the Home page	Need network access, need browser	1. Enter into the website A Novel Method For Handwritten Digit	The home page must be displayed properly	Working as expected	Pass

					Recognition System. 2. The instruction will show about how to use this website. 3. The next button will showto navigate to the next page(Recognition)			
HP_TC_OO2	UI	Home page	Check if the UI elemen ts are displayed properly in different screen sizes	Need network access, need browser	1. Enter into the website A Novel Method For Handwritten Digit Recognition System. 2. The instruction will show about how to use this website.	The home page must be displayed properly	Working as expected	Pass

					3. The next button will showto navigate to the next page(Recognition)			
HP_TC_OO3	Functional	Home page	Check if the next button in the Home page is working properly	Need network access, need browser	1. Enter into the website A Novel Method For Handwritten Digit Recognition System. 2. The instruction will show about how to use this website. 3. By clicking next button we can navigate to the next page(Recognition)	The next button will work properly, using this we can navigate to the next page	Working as expected	Pass
HP_TC_O		Recogniti	Verify UI	Need	1. Using	The	Working	

O4	UI	on page	elements in the Home page	network access, need browser, handwritten digit image in the local desk.	next button in the home page navigate to the recognition page. 2. Using choose file option select a handwritten digit image in the local desktop. 3. Using predict image button we can submit our input it will navigate to the next page and shows the predicted result.	choose file button and predict image button must display properly.	as expected	Pass
HP_TC_O O5	Functional	Recognition page	Verify the choose file option work	Need network access, need browser, handwrit	1. Using next button in the home page	Using choose file option we can select	Working as expected	Pass

			properly	ten digit image in the local desk.	navigate to the recogniti on page. 2. Using choose file option select a handwrit ten digit image in the local desktop. 3. Using predict image button we can submit our input it will navigate to the next page and shows the predict ed result.	input image from the local desk.		
HP_TC_O O6	Function al	Recogniti on page	Verify the predict image button work properly.	Need network access, need browser, handwrit ten digit image must be selected from the	1. Using next button in the home page navigate to the recogniti on page.	By clicking predict image button we can navigate to the predict ed result	Working as expected	Pass

				local system.	2. Using choose file option select a handwritten digit image in the local desktop. 3. Using predict image button we can submit our input it will navigate to the next page and shows the predicted result.	page.		
HP_TC_OO7	UI	Predicted result.	Verify the predicted result image is shown properly in the graph.	Need network access, need browser, handwritten digit image must be selected from the local system.	1. Click the predict image button in the recognition page. 2. It will navigate to the predicted result page	The predicted result page Must be declared properly.	Working as expected	Pass

					3. The predicted digit is show in the graph like structure. 4. In the Navbar there is two option Home and recognition using this button we will navigate to the respective page .			
HP_TC_OO8	Functional	Predicted result.	Verify the navigation button present in the page work properly	Need network access, need browser, handwritten digit image must be selected from the local system.	1. The Home button present in the page can used to navigate to the Home page 2. The Recognition button present	Home and Recognition button present in the page must work properly.	Working as expected	Pass

					in the page can used to navigate to the Recognition page.			

8.2 USER ACCEPTANCE TESTING

8.2.1. DEFFECT ANALYSIS

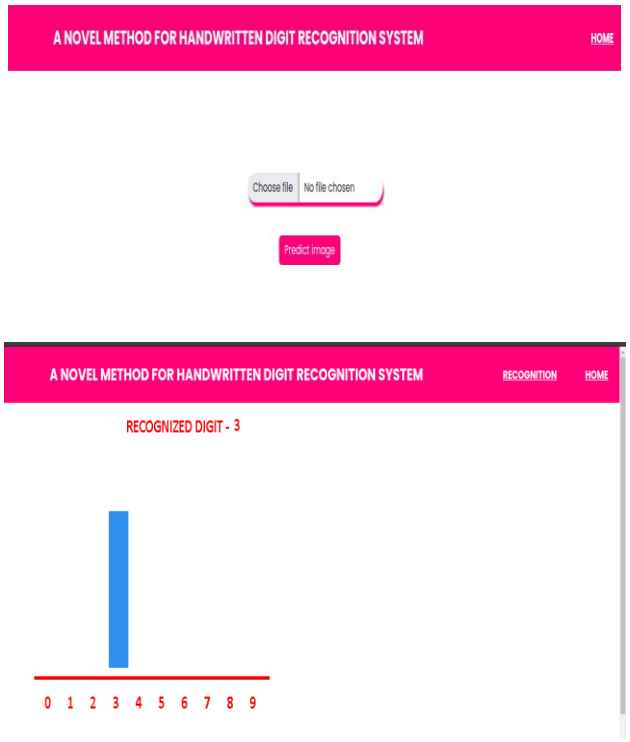
Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Total
By Design	1	0	1	0	2
Duplicate	0	0	0	0	0
External	0	0	2	0	2
Fixed	4	1	0	1	6
Not					
Reproduced	0	0	0	1	1
Skipped	0	0	0	1	1
won't Fix	1	0	1	0	2
Total	6	1	4	3	14

8.2.2 **TEST CASE ANALYSIS**

Section	TotalTested	NotTested	Fail	Pass
PrintEngine	0	0	0	0
Client Application	10	0	3	7
Security	2	0	1	1
Outsource shipping	0	0	0	0
Exception Reporting	2	0	0	2
Final report output	0	0	0	0
Version control	0	0	0	0

CHAPTER 9

PERFORMANCE METRICS

S.No.	Parameter	Values	Screenshot
1	Model summary	---	
2	Accuracy	Training Accuracy - 95% Validation Accuracy-88%	

CHAPTER 10

ADVANTAGES & DISADVANTAGES

ADVANTAGES

- Reduces manual work
- More accurate than average human

Capable of handling a lot of data

- Can be used anywhere from any device

DISADVANTAGES

- Cannot handle complex data
- All the data must be in digital format
- Requires a high performance server for faster predictions.
- Prone to occasional errors

CHAPTER 11

CONCLUSION

This project demonstrated a web application that uses machine learning to recognize handwritten numbers. Flask, HTML, CSS and a few other technologies were used to create this project. The model predicts the handwritten digit using a CNN network. During testing, the model achieved a 99.61% recognition rate. The proposed project is scalable and can easily handle a huge number of users. Since it is a web application, it is compatible with any device that can run a browser. This project is extremely useful in real-world scenarios such as recognizing number plates of vehicles, processing bank cheque amounts, numeric entries in forms filled up by hand (tax forms) and so on. There is so much room for improvement, which can be implemented in subsequent versions.

CHAPTER 12

FUTURE SCOPE

This project is far from complete and there is a lot of room for improvement. Some ofthe improvements that can be made to this project are as follows:

- Add support to detect from digits multiple images and save the results
- Add support to detect multiple digits
- Improve model to detect digits from complex images
- Add support to different languages to help users from all over the world

This project has endless potential and can always be enhanced to become better.

Implementing this concept in the real world will benefit several industries and reduce theworkload on many workers, enhancing overall work efficiency.

APPENDIX

SOURCE CODE

HTML PAGE

Index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Home page</title>
    <!-- Link External Stylesheet "style.css" -->
    <link rel="stylesheet" href="{{url_for('static',filename='css/style.css')}}">
    <!-- Link CSS Bootstrap Stylesheet -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css" rel="stylesheet">

</head>

<body>

<section id="home">
<!--Using id_name(home), access this section in style.CSS file-->
    <h1>A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM</h1>
</section>
```

```
<div class ="con-body">
    <!-- The con-body class is used to access a container portion.-->
    <div class = "instruction">
        <h1>INSTRUCTION</h1>
            <p>1.On the home page, you have a next button. You will click the next button to navigate to the recognition
                page.<br><br>
                2.On the recognition page, you will have an upload option where you can upload the digit image in the
                format of a scanned image or a document.<br><br>
                3.After uploading the image, click the "recognition" button. Then you will get a predicted result.<br><br
                4.On this page you have an option that gives the result in voice mode.<br><br>
                5.You have another option to continue the recognition. If you click the continue button, it will
                redirect to the recognition page.</p><br>

                <a href="{{ url_for('main') }}"><button type = "button">NEXT<img
                    src="{{url_for('static',filename='image/arrow.png')}}"></button></a>
    </div>
    <div class="con-2i">
        <img src="{{url_for('static',filename='image/bodyimg.jpeg')}}" alt="number image" class="num_img">
        <div class="color-box"></div>
    </div>

</div>
    <!-- JavaScript Bundle with Popper from bootstrap-->
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-OERcA2EqjJCMA+/3y+gxIOqMEjwtxJY7qPCqsdltbNJuaOe923+mo//f6V8Qbsw3" crossorigin="anonymous"></script>
</body>
```

main.html

```
<!DOCTYPE html>

<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <!--Title of this page is Recognition-->
    <title>Recognition</title>
    <!--Link external stylesheet styles.css present in static file-->
    <link rel="stylesheet" href="{{url_for('static',filename='css/styles.css')}}">
    <!-- Link CSS Bootstrap Stylesheet -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css" rel="stylesheet">

  </head>
  <body>
    <nav>
      <label class="logo">A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM</label>
      <ul>
       <!--Using this Home button in navbar we can return to home page-->
        <li><a class="active" href="{{ url_for('home') }}">Home</a></li>
      </ul>
    </nav>
    <br>
```

```
<div class = "col-md-offset-3 col-md-3" >
    <!--handwritten image in your localdesk can be upload using choose file option present in the recognition page-->
    <form action ="{{url_for('upload')}}" method="POST" enctype="multipart/form-data">
      <!-- The above upload function call the main.html page-->
      <input class="form-control" id="choose_file" type="file" name="imagefile"><br>
      <input class="btn btn-primary mt-3" id="new" type="submit" value="Predict image">
    </form>


</div>


</body>
</html>
```

Zero.html

```
<!DOCTYPE html>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Predicted result</title>
    <!--Link external stylesheet styles.css present in static file-->
    <link rel="stylesheet" href="{{url_for('static',filename='css/styles.css')}}">
    <!-- Link CSS Bootstrap Stylesheet -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css" rel="stylesheet">

</head>
<body>
    <nav>
        <label class="logo">A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM</label>
        <ul>
            <!--Using this Home button in navbar we can return to home page-->
            <!--Using this Recognition button in navbar we can return to recognition page-->
            <!--This home function call index.html using app.py-->
            <li><a class="active" href="{{ url_for('home') }}">Home</a></li>
        </ul>
       <ul>
            <!--This main function call main.html using app.py-->
            <li><a class="active" href="{{ url_for('main') }}">Recognition</a></li>
```

```
</head>
<body>
    <nav>
        <label class="logo">A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM</label>
        <ul>
            <!--Using this Home button in navbar we can return to home page-->
            <!--Using this Recognition button in navbar we can return to recognition page-->
            <!--This home function call index.html using app.py-->
            <li><a class="active" href="{{ url_for('home') }}">Home</a></li>
        </ul>
       <ul>
            <!--This main function call main.html using app.py-->
            <li><a class="active" href="{{ url_for('main') }}">Recognition</a></li>
        </ul>
    </nav>
<div class = "image">
    <!--This zero result is show if the  predicted value is zero-->
    <img src="{{url_for('static',filename='image/ZERO.png')}}" alt="number image" class="num_img">

</div>
</body>
</html>
```

one.html

```
<!DOCTYPE html>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Predicted result</title>
    <!--Link external stylesheet styles.css present in static file-->
    <link rel="stylesheet" href="{{url_for('static',filename='css/styles.css')}}">
    <!-- Link CSS Bootstrap Stylesheet -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css" rel="stylesheet">

</head>
<body>
    <nav>
        <label class="logo">A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM</label>
        <ul>
            <!--Using this Home button in navbar we can return to home page-->
            <!--Using this Recognition button in navbar we can return to recognition page-->
            <!--This home function call index.html using app.py-->
            <li><a class="active" href="{{ url_for('home') }}">Home</a></li>
        </ul>
       <ul>
            <!--This main function call main.html using app.py-->
            <li><a class="active" href="{{ url_for('main') }}">Recognition</a></li>
```

```
</head>
<body>
    <nav>
        <label class="logo">A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM</label>
        <ul>
            <!--Using this Home button in navbar we can return to home page-->
            <!--Using this Recognition button in navbar we can return to recognition page-->
            <!--This home function call index.html using app.py-->
            <li><a class="active" href="{{ url_for('home') }}">Home</a></li>
        </ul>
       <ul>
            <!--This main function call main.html using app.py-->
            <li><a class="active" href="{{ url_for('main') }}">Recognition</a></li>
        </ul>
    </nav>
<div class = "image">
    <!--This one result is show if the  predicted value is one-->
    <img src="{{url_for('static',filename='image/ONEpng.png')}}" alt="number image" class="num_img">

</div>
</body>
</html>
```

Two.html

```html
<!DOCTYPE html>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Predicted result</title>
    <!--Link external stylesheet styles.css present in static file-->
    <link rel="stylesheet" href="{{url_for('static',filename='css/styles.css')}}">
    <!-- Link CSS Bootstrap Stylesheet -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css" rel="stylesheet">

</head>
<body>
    <nav>
        <label class="logo">A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM</label>
        <ul>
            <!--Using this Home button in navbar we can return to home page-->
            <!--Using this Recognition button in navbar we can return to recognition page-->
            <!--This home function call index.html using app.py-->
            <li><a class="active" href="{{ url_for('home') }}">Home</a></li>
        </ul>
       <ul>
            <!--This main function call main.html using app.py-->
            <li><a class="active" href="{{ url_for('main') }}">Recognition</a></li>
```

```html
</head>
<body>
    <nav>
        <label class="logo">A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM</label>
        <ul>
            <!--Using this Home button in navbar we can return to home page-->
            <!--Using this Recognition button in navbar we can return to recognition page-->
            <!--This home function call index.html using app.py-->
            <li><a class="active" href="{{ url_for('home') }}">Home</a></li>
        </ul>
       <ul>
            <!--This main function call main.html using app.py-->
            <li><a class="active" href="{{ url_for('main') }}">Recognition</a></li>
        </ul>
    </nav>
<div class = "image">
    <!--This two result is show if the  predicted value is two-->
    <img src="{{url_for('static',filename='image/TWO.png')}}" alt="number image" class="num_img">

</div>
</body>
</html>
```

Three.html

```
<!DOCTYPE html>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Predicted result</title>
    <!--Link external stylesheet styles.css present in static file-->
    <link rel="stylesheet" href="{{url_for('static',filename='css/styles.css')}}">
    <!-- Link CSS Bootstrap Stylesheet -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css" rel="stylesheet">

</head>
<body>
    <nav>
        <label class="logo">A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM</label>
        <ul>
            <!--Using this Home button in navbar we can return to home page-->
            <!--Using this Recognition button in navbar we can return to recognition page-->
            <!--This home function call index.html using app.py-->
            <li><a class="active" href="{{ url_for('home') }}">Home</a></li>
        </ul>
       <ul>
            <!--This main function call main.html using app.py-->
            <li><a class="active" href="{{ url_for('main') }}">Recognition</a></li>
```

```
</head>
<body>
    <nav>
        <label class="logo">A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM</label>
        <ul>
            <!--Using this Home button in navbar we can return to home page-->
            <!--Using this Recognition button in navbar we can return to recognition page-->
            <!--This home function call index.html using app.py-->
            <li><a class="active" href="{{ url_for('home') }}">Home</a></li>
        </ul>
       <ul>
            <!--This main function call main.html using app.py-->
            <li><a class="active" href="{{ url_for('main') }}">Recognition</a></li>
        </ul>
    </nav>
<div class = "image">
    <!--This three result is show if the  predicted value is three-->
    <img src="{{url_for('static',filename='image/THREE.png')}}" alt="number image" class="num_img">

</div>
</body>
</html>
```

Four.html

```
<!DOCTYPE html>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Predicted result</title>
    <!--Link external stylesheet styles.css present in static file-->
    <link rel="stylesheet" href="{{url_for('static',filename='css/styles.css')}}">
    <!-- Link CSS Bootstrap Stylesheet -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css" rel="stylesheet">

</head>
<body>
    <nav>
        <label class="logo">A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM</label>
        <ul>
            <!--Using this Home button in navbar we can return to home page-->
            <!--Using this Recognition button in navbar we can return to recognition page-->
            <!--This home function call index.html using app.py-->
            <li><a class="active" href="{{ url_for('home') }}">Home</a></li>
        </ul>
       <ul>
            <!--This main function call main.html using app.py-->
            <li><a class="active" href="{{ url_for('main') }}">Recognition</a></li>
```

```
</head>
<body>
    <nav>
        <label class="logo">A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM</label>
        <ul>
            <!--Using this Home button in navbar we can return to home page-->
            <!--Using this Recognition button in navbar we can return to recognition page-->
            <!--This home function call index.html using app.py-->
            <li><a class="active" href="{{ url_for('home') }}">Home</a></li>
        </ul>
       <ul>
            <!--This main function call main.html using app.py-->
            <li><a class="active" href="{{ url_for('main') }}">Recognition</a></li>
        </ul>
    </nav>
<div class = "image">
    <!--This four result is show if the  predicted value is four-->
    <img src="{{url_for('static',filename='image/FOUR.png')}}" alt="number image" class="num_img">

</div>
</body>
</html>
```

Five.html

```
<!DOCTYPE html>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Predicted result</title>
    <!--Link external stylesheet styles.css present in static file-->
    <link rel="stylesheet" href="{{url_for('static',filename='css/styles.css')}}">
    <!-- Link CSS Bootstrap Stylesheet -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css" rel="stylesheet">

</head>
<body>
    <nav>
        <label class="logo">A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM</label>
        <ul>
            <!--Using this Home button in navbar we can return to home page-->
            <!--Using this Recognition button in navbar we can return to recognition page-->
            <!--This home function call index.html using app.py-->
            <li><a class="active" href="{{ url_for('home') }}">Home</a></li>
        </ul>
       <ul>
            <!--This main function call main.html using app.py-->
            <li><a class="active" href="{{ url_for('main') }}">Recognition</a></li>
```

```
</head>
<body>
    <nav>
        <label class="logo">A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM</label>
        <ul>
            <!--Using this Home button in navbar we can return to home page-->
            <!--Using this Recognition button in navbar we can return to recognition page-->
            <!--This home function call index.html using app.py-->
            <li><a class="active" href="{{ url_for('home') }}">Home</a></li>
        </ul>
       <ul>
            <!--This main function call main.html using app.py-->
            <li><a class="active" href="{{ url_for('main') }}">Recognition</a></li>
        </ul>
    </nav>
<div class = "image">
    <!--This five result is show if the  predicted value is five-->
    <img src="{{url_for('static',filename='image/FIVE.png')}}" alt="number image" class="num_img">

</div>
</body>
</html>
```

Six.html

```
<!DOCTYPE html>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Predicted result</title>
    <!--Link external stylesheet styles.css present in static file-->
    <link rel="stylesheet" href="{{url_for('static',filename='css/styles.css')}}">
    <!-- Link CSS Bootstrap Stylesheet -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css" rel="stylesheet">

</head>
<body>
    <nav>
        <label class="logo">A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM</label>
        <ul>
            <!--Using this Home button in navbar we can return to home page-->
            <!--Using this Recognition button in navbar we can return to recognition page-->
            <!--This home function call index.html using app.py-->
            <li><a class="active" href="{{ url_for('home') }}">Home</a></li>
        </ul>
       <ul>
            <!--This main function call main.html using app.py-->
            <li><a class="active" href="{{ url_for('main') }}">Recognition</a></li>
```

```
</head>
<body>
    <nav>
        <label class="logo">A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM</label>
        <ul>
            <!--Using this Home button in navbar we can return to home page-->
            <!--Using this Recognition button in navbar we can return to recognition page-->
            <!--This home function call index.html using app.py-->
            <li><a class="active" href="{{ url_for('home') }}">Home</a></li>
        </ul>
       <ul>
            <!--This main function call main.html using app.py-->
            <li><a class="active" href="{{ url_for('main') }}">Recognition</a></li>
        </ul>
    </nav>
<div class = "image">
    <!--This six result is show if the  predicted value is six-->
    <img src="{{url_for('static',filename='image/SIX.png')}}" alt="number image" class="num_img">

</div>
</body>
</html>
```

Seven.html

```
<!DOCTYPE html>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Predicted result</title>
    <!--Link external stylesheet styles.css present in static file-->
    <link rel="stylesheet" href="{{url_for('static',filename='css/styles.css')}}">
    <!-- Link CSS Bootstrap Stylesheet -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css" rel="stylesheet">

</head>
<body>
    <nav>
        <label class="logo">A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM</label>
        <ul>
            <!--Using this Home button in navbar we can return to home page-->
            <!--Using this Recognition button in navbar we can return to recognition page-->
            <!--This home function call index.html using app.py-->
            <li><a class="active" href="{{ url_for('home') }}">Home</a></li>
        </ul>
       <ul>
            <!--This main function call main.html using app.py-->
            <li><a class="active" href="{{ url_for('main') }}">Recognition</a></li>
```

```
</head>
<body>
    <nav>
        <label class="logo">A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM</label>
        <ul>
            <!--Using this Home button in navbar we can return to home page-->
            <!--Using this Recognition button in navbar we can return to recognition page-->
            <!--This home function call index.html using app.py-->
            <li><a class="active" href="{{ url_for('home') }}">Home</a></li>
        </ul>
       <ul>
            <!--This main function call main.html using app.py-->
            <li><a class="active" href="{{ url_for('main') }}">Recognition</a></li>
        </ul>
    </nav>
<div class = "image">
    <!--This seven result is show if the  predicted value is seven-->
    <img src="{{url_for('static',filename='image/SEVEN.png')}}" alt="number image" class="num_img">

</div>
</body>
</html>
```

Eight.html

```
<!DOCTYPE html>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Predicted result</title>
    <!--Link external stylesheet styles.css present in static file-->
    <link rel="stylesheet" href="{{url_for('static',filename='css/styles.css')}}">
    <!-- Link CSS Bootstrap Stylesheet -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css" rel="stylesheet">

</head>
<body>
    <nav>
        <label class="logo">A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM</label>
        <ul>
            <!--Using this Home button in navbar we can return to home page-->
            <!--Using this Recognition button in navbar we can return to recognition page-->
            <!--This home function call index.html using app.py-->
            <li><a class="active" href="{{ url_for('home') }}">Home</a></li>
        </ul>
       <ul>
            <!--This main function call main.html using app.py-->
            <li><a class="active" href="{{ url_for('main') }}">Recognition</a></li>
```

```
</head>
<body>
    <nav>
        <label class="logo">A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM</label>
        <ul>
            <!--Using this Home button in navbar we can return to home page-->
            <!--Using this Recognition button in navbar we can return to recognition page-->
            <!--This home function call index.html using app.py-->
            <li><a class="active" href="{{ url_for('home') }}">Home</a></li>
        </ul>
       <ul>
            <!--This main function call main.html using app.py-->
            <li><a class="active" href="{{ url_for('main') }}">Recognition</a></li>
        </ul>
    </nav>
<div class = "image">
    <!--This eight result is show if the  predicted value is eight-->
    <img src="{{url_for('static',filename='image/EIGHT.png')}}" alt="number image" class="num_img">

</div>
</body>
</html>
```

Nine.html

```
<!DOCTYPE html>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Predicted result</title>
    <!--Link external stylesheet styles.css present in static file-->
    <link rel="stylesheet" href="{{url_for('static',filename='css/styles.css')}}">
    <!-- Link CSS Bootstrap Stylesheet -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css" rel="stylesheet">

</head>
<body>
    <nav>
        <label class="logo">A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM</label>
        <ul>
            <!--Using this Home button in navbar we can return to home page-->
            <!--Using this Recognition button in navbar we can return to recognition page-->
            <!--This home function call index.html using app.py-->
            <li><a class="active" href="{{ url_for('home') }}">Home</a></li>
        </ul>
       <ul>
            <!--This main function call main.html using app.py-->
            <li><a class="active" href="{{ url_for('main') }}">Recognition</a></li>
```

```
</head>
<body>
    <nav>
        <label class="logo">A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM</label>
        <ul>
            <!--Using this Home button in navbar we can return to home page-->
            <!--Using this Recognition button in navbar we can return to recognition page-->
            <!--This home function call index.html using app.py-->
            <li><a class="active" href="{{ url_for('home') }}">Home</a></li>
        </ul>
       <ul>
            <!--This main function call main.html using app.py-->
            <li><a class="active" href="{{ url_for('main') }}">Recognition</a></li>
        </ul>
    </nav>
<div class = "image">
    <!--This nine result is show if the  predicted value is nine-->
    <img src="{{url_for('static',filename='image/NINE.png')}}" alt="number image" class="num_img">

</div>
</body>
</html>
```

STYLESHEETS

style.css

```css
/*This is the stylesheet for index.html*/
/*import google font poppins*/
@import url('https://fonts.googleapis.com/css2?family=Poppins:wght@300&display=swap');
*{
    margin: 0;
    padding: 0;
    box-sizing: border-box;
    font-family: 'Poppins', sans-serif;
}
.instruction{
    /* It is used to style content in the container. */
    flex-basis: 50%;
    position: relative;
    margin-left: 10%;

}

.con-body{
/* It used to style container in the body portion*/
display: flex;
justify-content: space-between;
align-items: center;
margin: 90px 0;
}
```

```css
.instruction h1{
    /*It is used to style the word "instruction." */
    font-size: 54px;
}
.instruction p{
    /*It styles the paragraph content present on the page.*/
    font-size: 20px;
    color: black;
    font-weight: 100;
    margin: 30px;

}
button{
    /* Style the next button */
    height: 40px;
    width: 140px;
    border: 0;
    padding: none;
    color: #fff;
    background: linear-gradient(to right, #fb5283, #ff3527);
    border-radius: 6px;
   transition: width 0.5s;
}
```

```css
button img{
    /*set the image for button*/
    width: 40px;
    display: none;
}


button:hover img{
    display: block;
}

button:hover{
    /*while clicking, set hover for the next button */
    width: 160px;
    display: flex;
    align-items: center;
    justify-content: space-between;
}
.instruction::after{
    /*Style for the leftmost vertical line*/
    content: '';
    width: 10px;
    height: 80%;
    background: linear-gradient(#fb5283,#ff3527);
```

```css
.instruction::after{
    /*Style for the leftmost vertical line*/
    content: '';
    width: 10px;
    height: 80%;
    background: linear-gradient(#fb5283,#ff3527);
    position: absolute;
    left:  -40px;
    top: 20px;
}
.con-2i{
    position: relative;
    flex-basis: 40%;
    display: flex;
    align-items:flex-end;
    margin: 40px;

}
.con-2i .num_img{
    height: 60%;
    width: 80%;


}
```

```css
.color-box{
    /*This code is used to create a green-colored box like structure.*/
    position: absolute;
    right: 45%;
    top: 10px;
    background: linear-gradient(#1dc280,#3b9cb7);
    border-radius: 10px 10px 10px 10px;
    height: 300px;
    width: 80%;
    z-index: -1;
    transform: translateX(200px);
    }

    #home{

        /*Style for the background image
        Set the width,height, background-size, and background-position of the image.
        */
        width: 100%;
        min-height: 100vh;
        display: flex;
        justify-content: center;
        align-items: center;
        /*linear-gradient-progressive transition between two or more colours in a straight line*/
        background: linear-gradient(rgba(82, 208, 240, 0.1),rgba(151, 16, 128, 0.1)),url('/static/image/background_im
```

```css
        width: 100%;
        min-height: 100vh;
        display: flex;
        justify-content: center;
        align-items: center;
        /*linear-gradient-progressive transition between two or more colours in a straight line*/
        background: linear-gradient(rgba(82, 208, 240, 0.1),rgba(151, 16, 128, 0.1)),url('/static/image/background_im
        background-size: cover;
        background-position: center;


    }

    #home h1{

    /* Style the title of the project
    Set the font size of the text as 70px.
    Align the text to the center.
    Set the text colour as white.
    Set the spacing between letters to 2px.
    Set the text shadow as white up to 6px.
    */
    font-size: 70px;
    text-align: center;
    color: white;
```

```
    #home h1{

    /* Style the title of the project
    Set the font size of the text as 70px.
    Align the text to the center.
    Set the text colour as white.
    Set the spacing between letters to 2px.
    Set the text shadow as white up to 6px.
    */
    font-size: 70px;
    text-align: center;
    color: white;
    letter-spacing: 2px;
    font-family: Cambria, Cochin, Georgia, Times, 'Times New Roman', serif;
    text-shadow: 0px 1px 0px #FFFFFF,
                 0px 2px 0px #FFFFFF,
                 0px 3px 0px #FFFFFF,
                 0px 4px 0px #FFFFFF,
                 0px 5px 0px #FFFFFF,
                 0px 6px 0px #FFFFFF;


}
```

styles.css

```
/*This is the stylesheet for main.html*/
/*import google font poppins*/
@import url('https://fonts.googleapis.com/css?family=Poppins:400,500,600,700&display=swap');
*{
  margin: 0;
  padding: 0;
  box-sizing: border-box;
  font-family: 'Poppins', sans-serif;
}
body{
    font-family: montserrat;
}
nav{
  /*It is used to set the color,height and width of the navbar*/
    background: #fb5283;
    height: 80px;
    width: 100%;
}
label.logo{
  /*It is used to style the text in the navbar*/
    color: white;
    font-size: 25px;
    line-height: 80px;
    padding: 0 100px;
    font-weight: bold;
}
```

```css
/*This three section nav ul,ul li, ul li a style the navigation button*/
nav ul{
    float:right;
    margin-right: 20px;
}
nav ul li{
    display: inline-block;
    line-height: 80px;
    margin: 0 5px;
}
nav ul li a{
    color:white;
    font-size: 17px;
    border-radius: 3px;
    text-transform: uppercase;
    font-weight: bold;
}
/*This portion style the choose file button*/
#choose_file{
  width: 300px;
  background: white;
  border: none;
  outline: none;
  box-shadow: 2px 5px 2px #fb5283;
  border-radius: 50px;
  position: relative;
```

```css
    outline: none;
    box-shadow: 2px 5px 2px #fb5283;
    border-radius: 50px;
    position: relative;
    top: 100px;
    left:530px;
  }

  /*This style the pedict image button*/
  #new{
    background: #fb5283;
    border: none;
    top: 100px;
    position: relative;
    left: 600px;
  }

  ::-webkit-file-upload-button{
    border: none;
    background: #fb5283;
  }
  ::-webkit-file-upload-button{
    border: none;
    background: #fb5283;
  }
```

PYTHON PAGE

app.py

```python
#import the packages
from flask import Flask, render_template,request
import tensorflow as tf
import numpy as np
from PIL import Image
from tensorflow.keras.models import load_model

app=Flask(__name__,template_folder='template')

#index page load
@app.route("/")
@app.route("/index")
def home():
    return render_template("index.html")

#load the  main page
@app.route("/main",methods=['GET'])
def main():
    return render_template("main.html")

# get the image in app.py and predict the image
@app.route("/main",methods=['POST'])
def upload():

# already trained model as been save this code load the model here
    model = load_model('model/mnistCNN.h5')
```

```python
#image is preprocceesing and predicte the image
    imagefile = request.files['imagefile']
    if imagefile:
        img = Image.open(request.files['imagefile'].stream).convert("L")
        img = img.resize((28,28))
        image = np.array(img)
        image = image.reshape(1,28,28,1)
        prid = model.predict(image)
        prid =np.argmax(prid,axis= 1)

# predict the image and if condition using load the html page
        if (int(prid) == 0):
            return render_template("zero.html")
        if (int(prid)== 1):
            return render_template("one.html")
        if (int(prid)== 2):
            return render_template("two.html")
        if (int(prid)== 3):
            return render_template("three.html")
        if (int(prid)== 4):
            return render_template("four.html")
        if (int(prid)== 5):
            return render_template("five.html")
        if (int(prid)== 6):
            return render_template("six.html")
        if (int(prid)== 7):
            return render_template("seven.html")
        if (int(prid)== 8):
            return render_template("eight.html")
        if (int(prid)== 9):
            return render_template("nine.html")
    else:
        return render_template("main.html")
if __name__=='__main__':
    app.run(debug=True)
```

GITHUB

https://github.com/IBM-EPBL/IBM-Project-23789-1659929348

PROJECT DEMO

https://github.com/IBM-EPBL/IBM-Project-23789-1659929348/blob/main/FINAL%20DELIVARABLES/Demonstration%20Video/A%20Novel%20Method%20for%20Handwritten%20Digit%20Recognition%20System.mp4

GITHUB

https://github.com/IBM-EPBL/IBM-Project-23789-1659929348

PROJECT DEMO

https://github.com/IBM-EPBL/IBM-Project-23789-1659929348/blob/main/FINAL%20DELIVARABLES/Demonstration%20Video/A%20Novel%20Method%20for%20Handwritten%20Digit%20Recognition%20System.mp4